Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники» Филиал  
«Минский радиотехнический колледж»

Учебная дисциплина «Программные средства создания Internet-приложений»

**Отчет**

по выполнению лабораторной работы

«Создание приложения с использованием асинхронных событий»

Выполнила Гончаров П.В.

Проверила Терешко О.И.

Минск 2020

Лабораторная работа № 29

Номер учебной группы: 7K2492.

Фамилия, инициалы учащегося: Гончаров П.В.

Дата выполнения работы: 05.03.2021

Тема работы: «Создание приложения с использованием асинхронных событий»

Цель работы: Формирование умений использования асинхронных событий в сценариях на языке JavaScript.

Результат выполнения работы:

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Lab 29</title>

  <script src="script.js"></script>

</head>

<body>

  <form>

    <textarea id="textarea1" style="width: 170px; height: 100px;"></textarea>

    <br>

    <button onclick="task1(event)">task 1</button>

  </form>

  <br>

  <span>Интервал</span>

  <input type="number" name="interval" id="interval" min="1">

  <button onclick="task2()">task 2</button>

  <div id="images">

    <img src="" id="image" alt="">

  </div>

</body>

</html>

function task1(event) {

  const textarea1 = document.getElementById("textarea1");

  event.preventDefault();

  setTimeout(() => {

    textarea1.dispatchEvent(new Event("focus"));

  }, 0);

  textarea1.dispatchEvent(new Event("click"));

}

window.onload = function(){

  textarea1.addEventListener("focus", () => {

    textarea1.value += "произошел фокус\n";

  });

  textarea1.addEventListener("click", () => {

    textarea1.value += "произошел клик\n";

  })

}

let arrImg = [

  "https://sun9-15.userapi.com/impg/lM-J8XmiTYg\_x63Cdfk32o\_EBCFytqoAZhGtog/\_PkYs758MHU.jpg?size=500x500&quality=96&sign=34a62e22fd07507f370f8bbc173d6f57&type=album",

  "https://sun9-18.userapi.com/impg/iFmixUPKnOr8HykOb-ZcEZUzsAbpu5UtmTY2Mw/r-HiF-m9zEs.jpg?size=500x500&quality=96&sign=0cda884f111bb01a7819d05e1e2b9762&type=album",

  "https://sun9-63.userapi.com/impg/neVcQT6sGm9bhK8fftmCtpYpHOO3y7WZuq-j5w/FqKTmQBySMw.jpg?size=500x500&quality=96&sign=16735a2245fe2bc5bbfc215feb3b41b6&type=album",

  "https://sun9-75.userapi.com/impg/ePkirK2zICtSokmNxVDsnlw68bSJI\_dB75RoLg/wxz9q3pAbcg.jpg?size=500x500&quality=96&sign=a1c2da2f984d9e9f8710c8c922b55688&type=album",

  "https://sun9-76.userapi.com/impg/0h9fbtScrSMB-uUULLr9j\_RiGf8pQAfBTVP0Pw/E8yO0Z5Q10I.jpg?size=500x500&quality=96&sign=8ff21d7a09dd618c598478176a1e3c7e&type=album"

];

let iNextImg = 0;

function task2() {

  const interval = Number(document.querySelector("#interval")?.value);

  if (!interval || interval <= 0) {

    alert("Ошибка! Введите итервал равный больше нуля");

    return;

  }

  getNextImg(interval);

}

function getNextImg(interval) {

  setTimeout(() => {

    fetch(arrImg[iNextImg % arrImg.length])//получаем картинку

      .then(response => {

        return response.blob();//возвращаем обеъкт blob - путь и последовательность др. объектов

      })

      .then(imgBlob => {

        const imageSrc = URL.createObjectURL(imgBlob);//создание ссылки на объект

        const imageElement = document.querySelector("#image");

        imageElement.src = imageSrc;

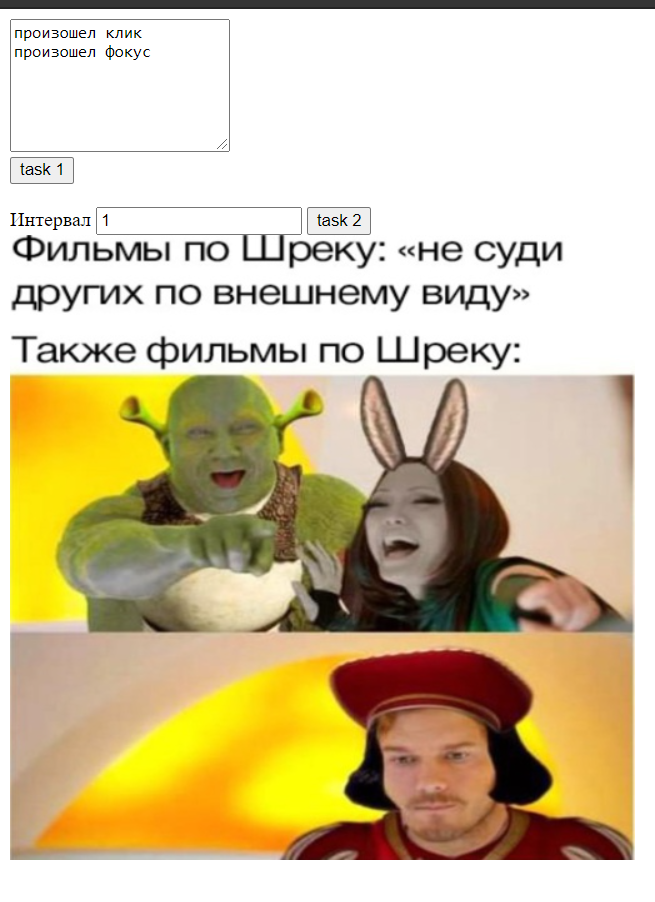
        iNextImg++;

        getNextImg(interval);

      })

  }, interval \* 1000)

}



Контрольные вопросы и задания

1. Что представляет собой объект Promise?

Объект используется для отложенных и асинхронных вычислений

1. Укажите внутренние свойства объекта Promise. Как они влияют на объект?
2. state («состояние») – вначале "pending" («ожидание»), потом меняется на "fulfilled" («выполнено успешно») при вызове resolve или на "rejected" («выполнено с ошибкой») при вызове reject;

2) result («результат») – вначале undefined, далее изменяется на value при вызове resolve(value) или на error при вызове reject(error).

3. Для чего предназначены методы .then/.catch/.finally?

Для обработки результата промиса использовать методы .then/.catch/.finally. Метод .then() возвращает Promise. Метод catch() возвращает Promise(обещание) и работает только в случае отклонения обещания. В случае вызова .finally(f) функция f выполнится в любом случае, когда промис завершится: успешно или с ошибкой.

1. В чем суть использования «async/await» при работе с промисами?

функция async  всегда возвращает промис